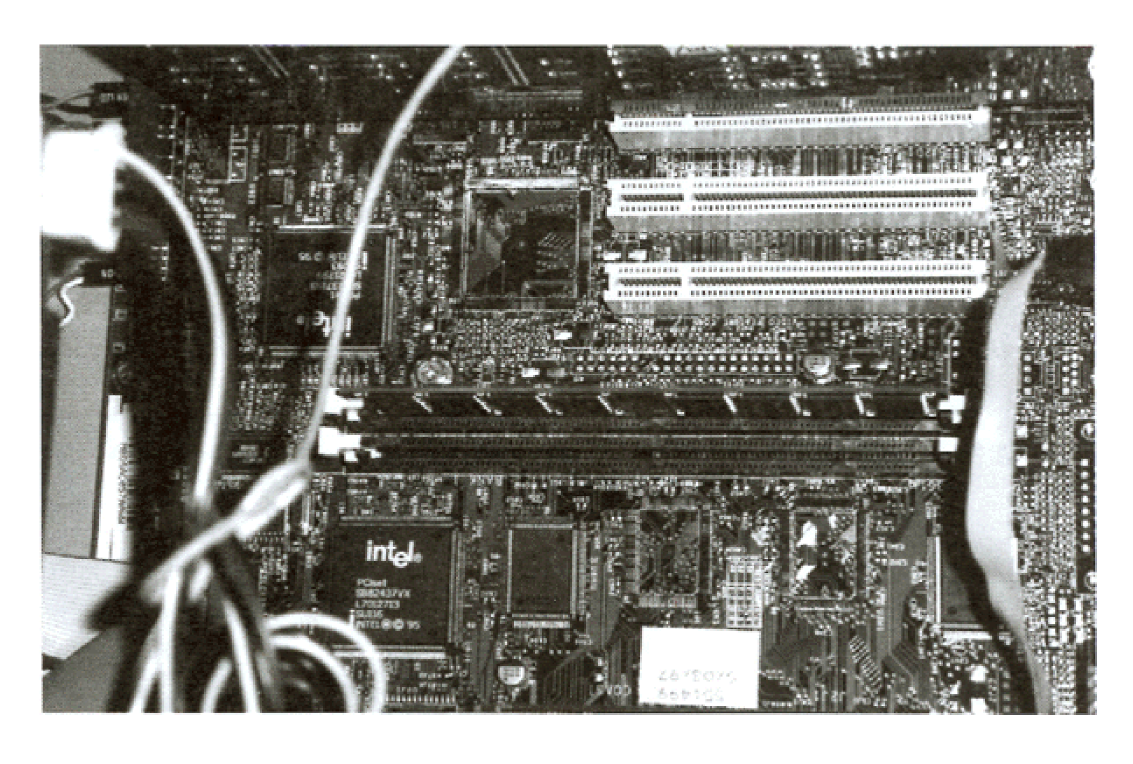




Module d'Electronique

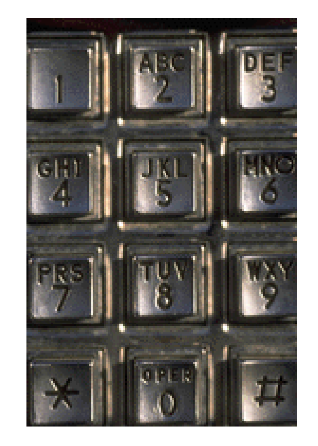
2ème partie : Electronique numérique

© Fabrice Sincère version 3.0.7

http://pagesperso-orange.fr/fabrice.sincere

Chapitre 1 Représentation des nombres

1-1- Numération dans le système décimal (base 10)



10 chiffres sont utilisés : 0 à 9.

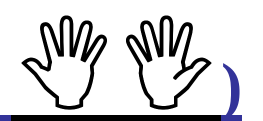
Un nombre se décompose de la façon suivante :

4792 = $4 \times 1000 + 7 \times 100 + 9 \times 10 + 2 \times 1$

1-2- Généralisation : numération en base B

Dans la base B, B chiffres sont utilisés.

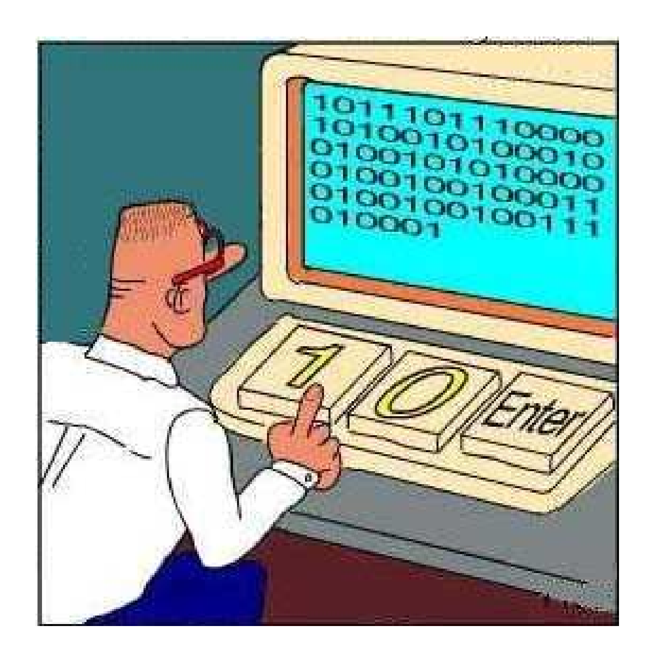
$(a_n a_{n-1} \ldots a_0)_B = a_n B^n + a_{n-1} B^{n-1} + \ldots + a_0 B^0$

1-2-1- B=10 : base décimale (utilisée par l'homme )

$(4792)_{10} = 4 \times 10^3 + 7 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$

1-2-2- B=2 : base binaire (utilisée par les systèmes numériques)

C’est la base la plus simple :
deux chiffres (ou ***bits : binary digits***) 0 et 1.



$$(10010011)_2 = 1\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$$
$$= 128 + 16 + 2 + 1$$
$$= (147)_{10}$$

Remarque : un ***byte*** (ou ***octet***) est une information de 8 bits.

1-3- Changement de base

- Passage du système décimal vers le système binaire

On décompose le nombre en puissance de 2 :

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 …

$(29)_{10} = 16 + 8 + 4 + 1 = (11101)_2$

- Remarque : Calculatrice (en mode scientifique) de Windows®

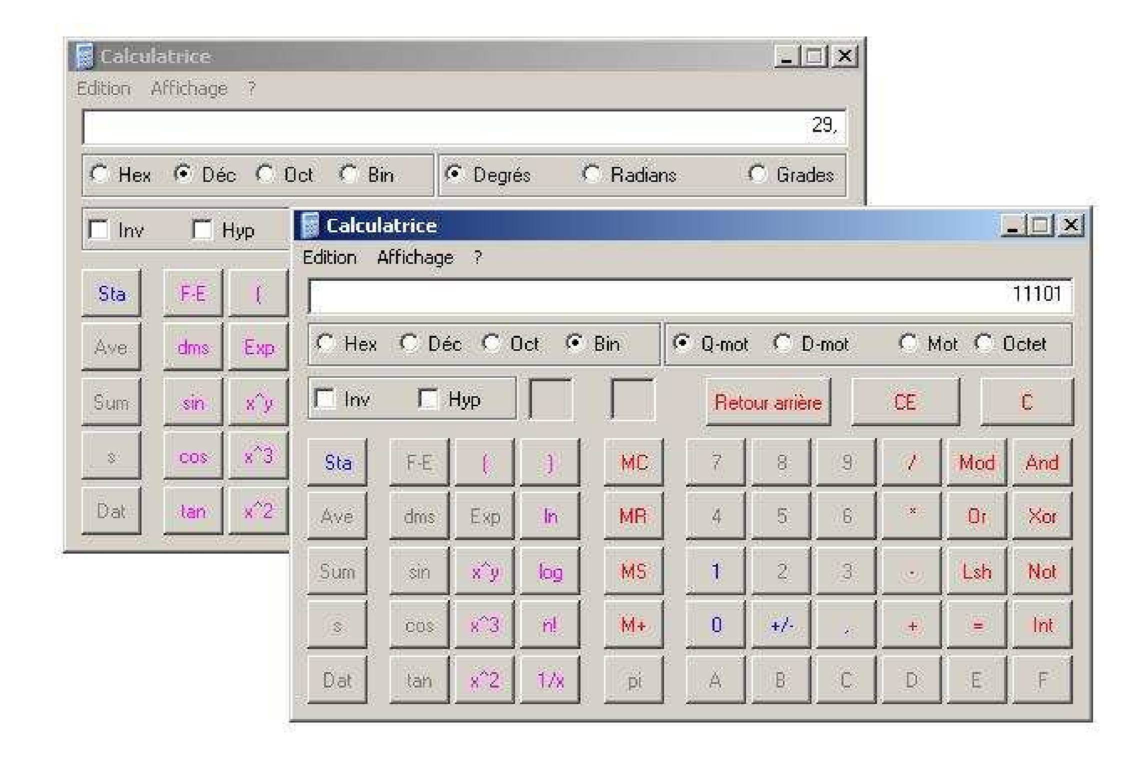


Table 1

Décimal (base 10)	Binaire naturel (base 2)	Octal (base 8)	Hexadécimal (base 16)
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
127	1111111	177	7F
128	10000000	200	80
255	11111111	377	FF
256	1 00000000	400	100

Chapitre 2 Fonctions logiques

2-1- Fonctions logiques de base

- Fonction logique ET

Je vais au cinéma ce soir si Alain **et** Bertrand viennent avec moi.

Table de vérité (table 2)

Alain	Bertrand	Sortie
Ne vient pas	Ne vient pas	Pas de cinéma
Ne vient pas	Vient	Pas de cinéma
Vient	Ne vient pas	Pas de cinéma
Vient	Vient	Cinéma

Variable logique : une variable logique peut prendre deux états.

Alain ne vient pas :	a = 0
Alain vient :	a = 1
Bertrand ne vient pas :	b = 0
Bertrand vient :	b = 1
Pas de sortie cinéma :	s = 0
Sortie cinéma :	s = 1

Table 3

a	b	s
0	0	0
0	1	0
1	0	0
1	1	1

Equation logique (ou équation booléenne) : $s = a \text{ ET } b$

On utilise le symbole · pour désigner la fonction logique ET
(ne pas confondre avec la multiplication) : $s = a \cdot b$

Autre écriture : $s = ab$

- Fonction logique OU

Je vais au cinéma ce soir si Alain **ou** Bertrand viennent avec moi.

Equation logique $s = a \text{ OU } b$

On utilise le symbole **+** pour désigner la fonction logique OU
(ne pas confondre avec l'addition) : $s = a + b$

Table de vérité (table 4)

a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

$1 + 1 = 1$

• Fonction logique NON

Je ne vais **pas** au cinéma ce soir si Emma vient.

Equation logique s = NON e

On utilise la *barre de complément* pour désigner la fonction NON :

$$s = \overline{e}$$

Table de vérité (table 5)

e	s
0	1
1	0

$\overline{0} = 1$

2-2- Fonctions logiques dérivées

• Fonction logique NON ET

Je ne vais **pas** au cinéma ce soir si Alain **et** Bertrand viennent.

Equation logique s = NON (a ET b)

$$s = \overline{a \cdot b} = \overline{ab}$$

Table de vérité (table 6)

a	b	s
0	0	1
0	1	1
1	0	1
1	1	0

$$\overline{0 \cdot 1} = \overline{0} = 1$$

- Fonction logique NON OU

Equation logique s = NON (a OU b)

$$s = \overline{a + b}$$

Table de vérité (table 7)

a	b	s
0	0	1
0	1	0
1	0	0
1	1	0

$$\overline{0 + 0} = \overline{0} = 1$$

- Fonction logique OU exclusif

Table de vérité (table 8)

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

s = 1 s'il y a un nombre impair d'entrées à l'état 1.

Equation logique $s = a \oplus b$

Le signe $\oplus$ désigne la fonction logique OU exclusif.

• Fonction logique NON OU exclusif

Equation logique $s = \overline{a \oplus b}$

Table de vérité (table 9)

a	b	s
0	0	1
0	1	0
1	0	0
1	1	1

$\overline{1 \oplus 1} = \overline{0} = 1$

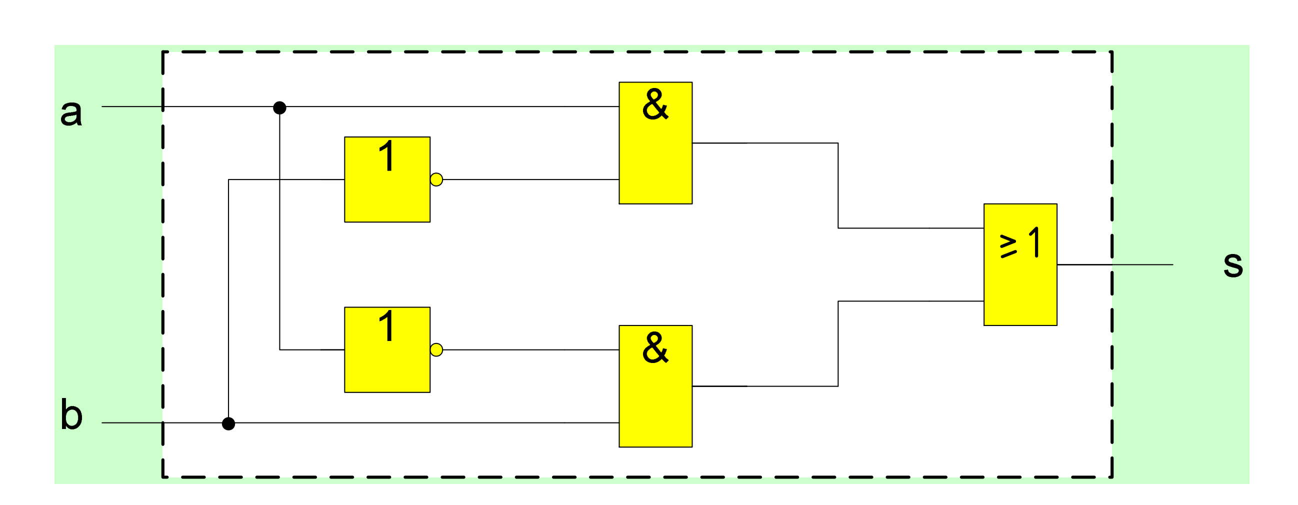
s = 1 s’il y a un nombre pair d’entrées à l’état 1.

2-3- Représentation symbolique (tableau 10)

Fonction logique	Symbole européen	Symbole américain	Table de vérité
OUI $s = e$	e 1 s		e s 0 0 1 1
NON (NO) $s = \overline{e}$	e 1 s		e s 0 1 1 0
ET (AND) $s = a \cdot b$	a & s b		a b s 0 0 0 0 1 0 1 0 0 1 1 1
OU (OR) $s = a + b$	a ≥ 1 s b		a b s 0 0 0 0 1 1 1 0 1 1 1 1
NON ET (NAND) $s = \overline{a \cdot b}$	a & s b		a b s 0 0 1 0 1 1 1 0 1 1 1 0
NON OU (NOR) $s = \overline{a + b}$	a ≥ 1 s b		a b s 0 0 1 0 1 0 1 0 0 1 1 0
OU exclusif (EXOR) $s = a \oplus b$	a =1 s b		a b s 0 0 0 0 1 1 1 0 1 1 1 0
NON OU exclusif (EXNOR) $s = \overline{a \oplus b}$	a =1 s b		a b s 0 0 1 0 1 0 1 0 0 1 1 1

2-4- Logigrammes

- Exemple (figure 1)



Equation booléenne de la sortie : $s = a\overline{b} + \overline{a}b$

Table de vérité (table 11)

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

$0 \cdot \overline{1} + \overline{0} \cdot 1 = 0 \cdot 0 + 1 \cdot 1 = 0 + 1 = 1$

Remarque : il s'agit aussi de la fonction OU exclusif : $a\overline{b} + \overline{a}b = a \oplus b$

2-5- Algèbre de Boole

2-5-1- Propriétés des fonctions logiques (tableau 12)

Propriétés		Théorèmes
Commutativité	$a + b = b + a$ $a \cdot b = b \cdot a$	
Associativité	$(a + b) + c = a + (b + c) = a + b + c$ $(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$	
Distributivité	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$ $a + (b \cdot c) = (a + b) \cdot (a + c)$	
Eléments neutres	$a + 0 = a$ $a \cdot 1 = a$	Des bornes universelles
Complémentation	$a + \overline{a} = 1$ $a \cdot \overline{a} = 0$	De non contradiction
	$a + a = a$ $a \cdot a = a$	D'idempotence $1 + 1 = 1$ $1 \cdot 1 = 1$
Eléments absorbants	$a \cdot 0 = 0$ $a + 1 = 1$	Du tiers exclus $1 \cdot 0 = 0$ $0 + 1 = 1$
	$a + (a \cdot b) = a + a \cdot b = a$ $a \cdot (a+b)=a$	D'absorption
	$\overline{\overline{a}} = a$	D'involution
	$a \cdot b + a \cdot \overline{b} = a$	D'inclusion
	$\overline{a + b + c} = \overline{a} \cdot \overline{b} \cdot \overline{c}$ $\overline{a \cdot b \cdot c} = \overline{a} + \overline{b} + \overline{c}$	De de Morgan
	$a \oplus (b \oplus c) = (a \oplus b) \oplus c = a \oplus b \oplus c$	
	$a \oplus a = 0$ $a \oplus \overline{a} = 1$	
	$a \oplus 0 = a$ $a \oplus 1 = \overline{a}$	

2-5-2- Application à la simplification ou transformation d'expressions logiques

- Exemple

$$s = \overline{a}\,b\,\overline{c} + a\,b\,\overline{c}$$

$$= (\overline{a} + a)\,b\,\overline{c}$$

$$= 1 \cdot b\,\overline{c}$$

$$s = b\,\overline{c}$$

Chapitre 3 Circuits intégrés logiques

En électronique, des C.I. spécialisés permettent de réaliser les fonctions logiques.
Il existe deux grandes familles de C.I. logiques.

3-1- Famille TTL

La famille TTL (Transistor Transistor Logic) est fabriquée avec des transistors bipolaires.

En logique TTL-standard :
Tension d'alimentation : Vcc = (5 ± 0,25) V

En entrée :	0 à 0,8 V :	niveau logique 0
	2 à 5 V :	niveau logique 1
En sortie :	0 à 0,4 V :	niveau logique 0
	2,4 à 5 V :	niveau logique 1

• Exemple : circuit intégré 7400

Ce circuit dispose de quatre fonctions (ou **portes**) logiques NON ET (NAND) à 2 entrées :

Fig. 2 : Brochage

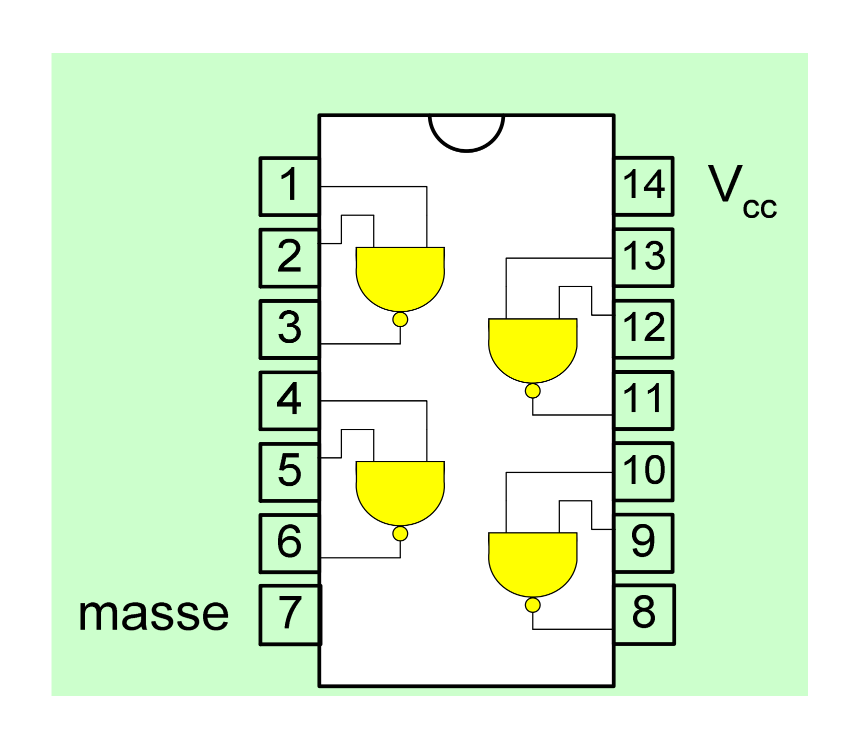
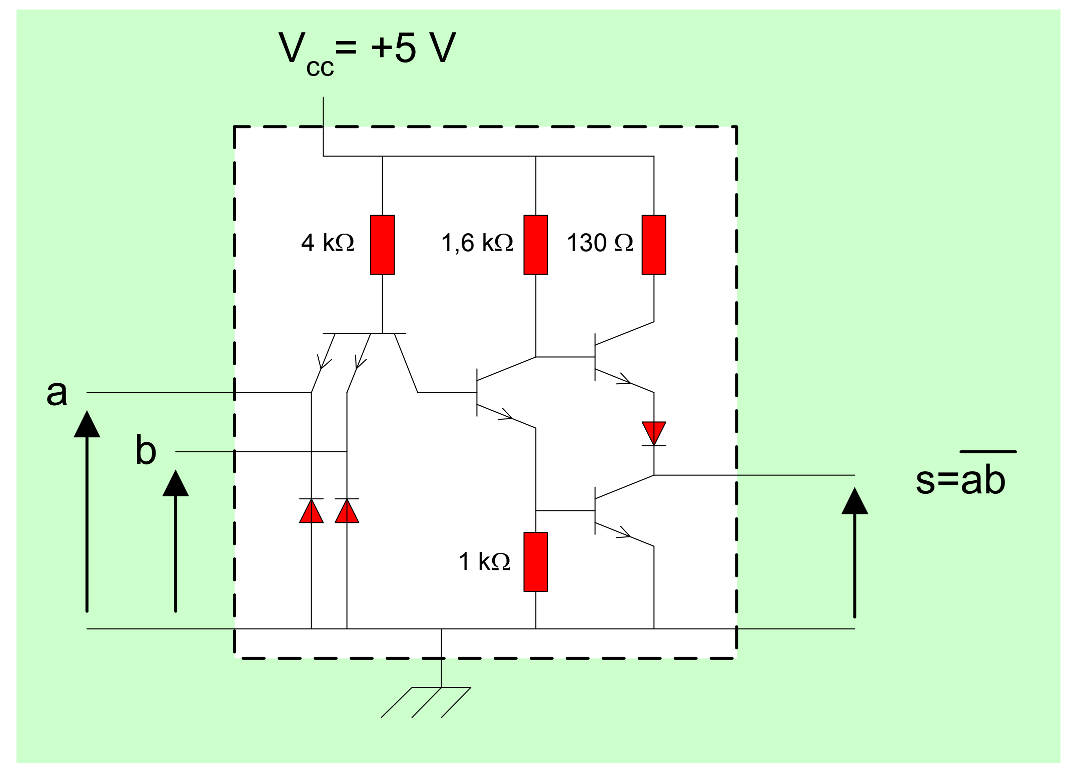


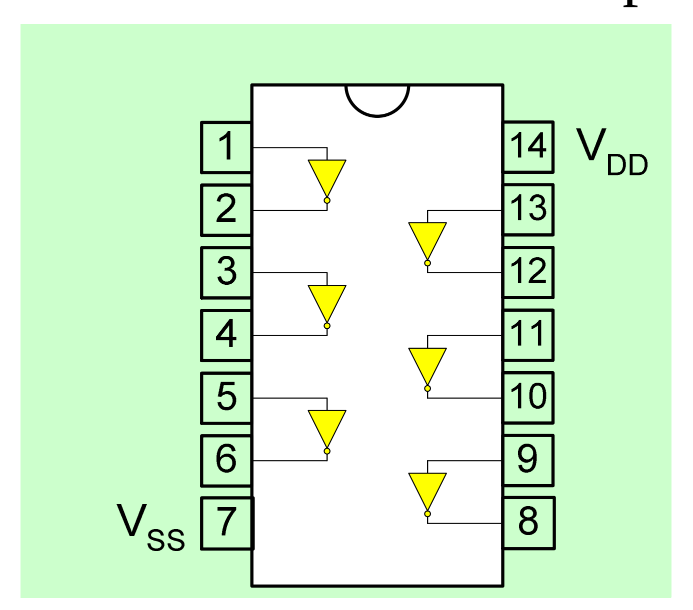
Fig. 3 : Schéma interne d'une porte NAND

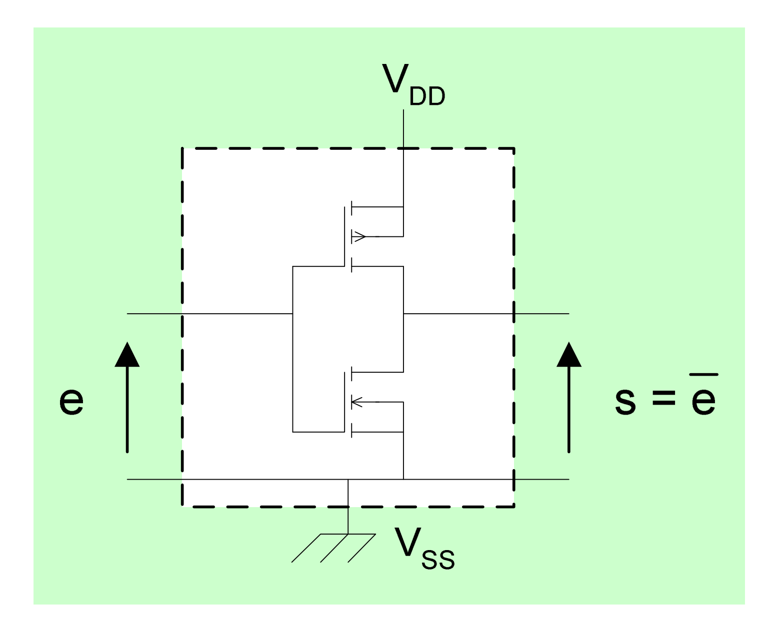


3-2- Famille CMOS

La famille CMOS (Complementary Metal Oxide Semiconductor) est fabriquée avec des transistors MOSFET.
Tension d'alimentation : 3 à 18 V

- Exemple : circuit intégré 4069B (fig. 4 et 5)
  Ce circuit contient six portes inverseuses NON :





Remarque : les familles CMOS et TTL ne sont pas compatibles ☹

Chapitre 4 Circuits combinatoires

Un circuit combinatoire est un circuit logique où chacune des sorties est une fonction logique des entrées.

4-1- Synthèse d'un circuit combinatoire

On désire avoir un circuit logique à trois entrées et deux sorties dont la table de vérité (table 13) est :

a	b	c	s_1	s_0
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

a	b	c	s_1	s_0
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	**0**	**0**	1	**1**
1	**0**	**1**	0	**1**
1	1	0	0	0
1	1	1	1	0

Equations logiques

s_0 = 1 si (a=1 **et** b=0 **et** c=0)

ou (a=1 **et** b=0 **et** c=1)

$$s_0 = a\,\bar{b}\,\bar{c} + a\,\bar{b}\,c$$

$$s_1 = \bar{a}\,\bar{b}\,\bar{c} + a\,\bar{b}\,\bar{c} + a\,b\,c$$

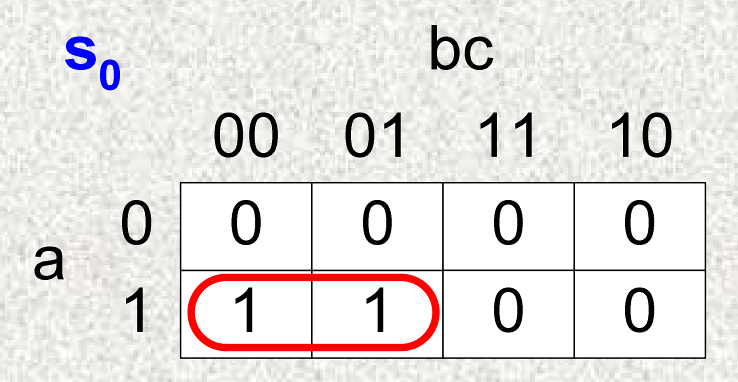
Simplification des équations logiques

$$s_0 = a\,\bar{b}\,\bar{c} + a\,\bar{b}\,c = a\,\bar{b}\,(\bar{c} + c) = a\,\bar{b}$$

$$s_0 = 1 \text{ si } (a = 1 \text{ et } b = 0)$$

$$s_1 = (\bar{a} + a)\bar{b}\,\bar{c} + a\,b\,c = \bar{b}\,\bar{c} + a\,b\,c$$

• Il est plus efficace d'utiliser la technique des « tableaux de Karnaugh » pour simplifier les équations logiques :



s_0

a \ bc	00	01	11	10
0	0	0	0	0
1	1	1	0	0

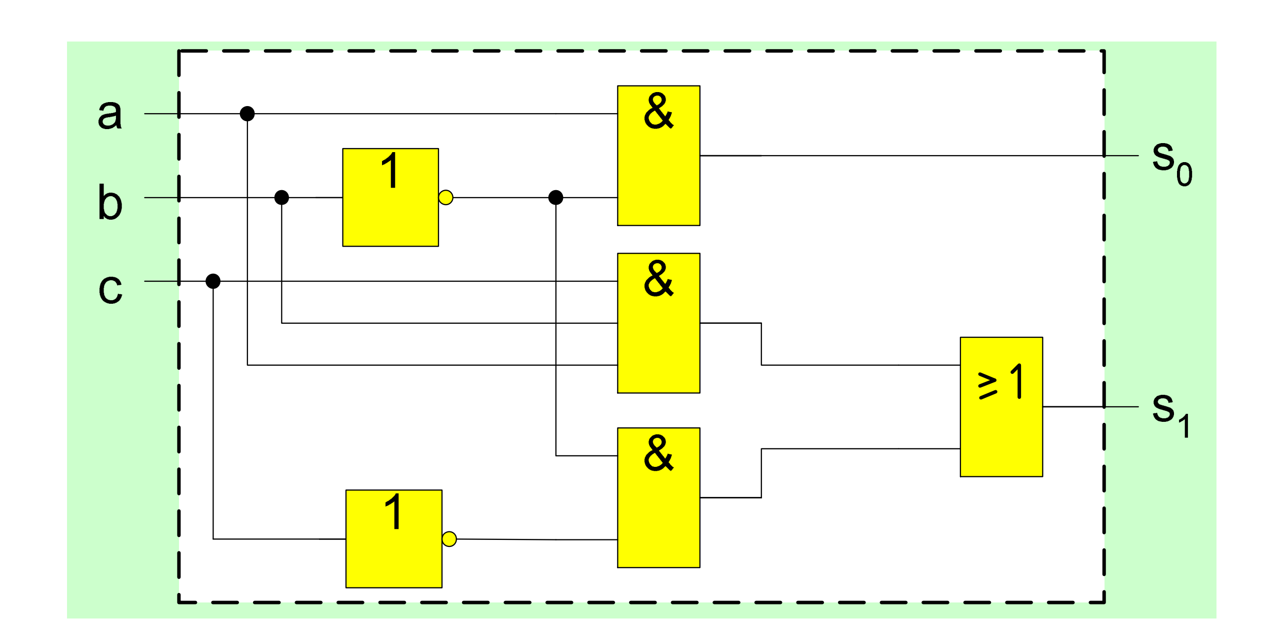
$$s_0 = a\,\overline{b}$$

s_1

a \ bc	00	01	11	10
0	1	0	0	0
1	1	0	1	0

$$s_1 = \overline{b}\,\overline{c} + a\,b\,c$$

Logigramme (fig. 6)



$$s_0 = a\,\overline{b}$$

$$s_1 = \overline{b}\,\overline{c} + a\,b\,c$$

4-2- Circuits arithmétiques

A partir de fonctions logiques, on peut créer les fonctions arithmétiques : addition, soustraction, multiplication, division, comparaison …
On utilise la base binaire.

4-2-1- Comparateur 1 bit

Ce circuit doit réaliser la comparaison de deux nombres binaires de un bit : a, b.
Le résultat de la comparaison est donné par l'état de la sortie :

- s = 1 si a = b
- s = 0 si a ≠ b

Table de vérité (table 14)

s = 1 si a = b

s = 0 si a ≠ b

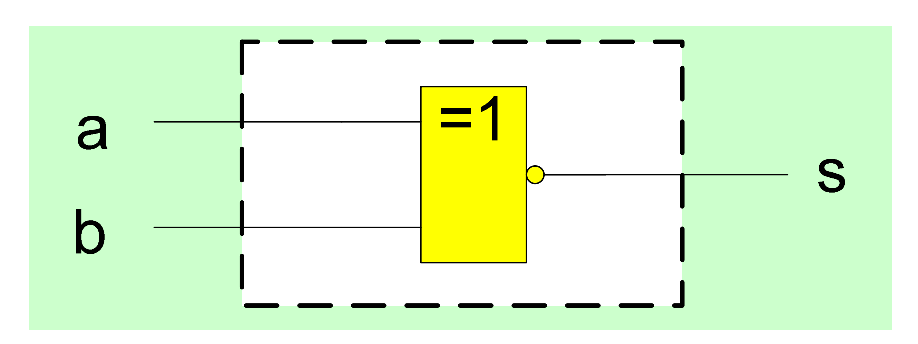
a	b	s
0	0	1
0	1	0
1	0	0
1	1	1

Equation booléenne de la sortie

$$s = \bar{a}\,\bar{b} + ab$$

$$s = \overline{a \oplus b}$$ (fonction NON OU exclusif)

Logigramme (fig. 7)



4-2-2- Additionneur 1 bit

Ce circuit doit réaliser l'addition arithmétique de deux nombres binaires de un bit : a, b.
Le résultat de l'addition nécessite un nombre de deux bits ($s_1 s_0$).

Table de vérité (table 15)

a	b	s_1(MSB)	s_0(LSB)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

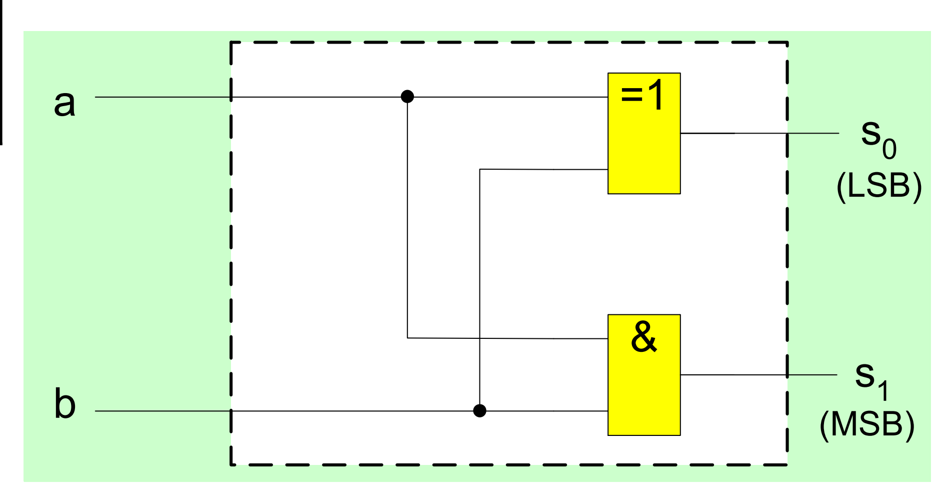
1 + 1 = 2 (en décimal)

1 + 1 = 10 (en binaire)

Equation booléenne des sorties

$$s_0 = a \oplus b$$
$$s_1 = ab$$

Logigramme (fig. 8)



Chapitre 5 Circuits séquentiels

Dans un circuit séquentiel, les sorties sont des fonctions logiques des entrées **et de l'état antérieur des sorties**.

5-1- Fonction mémoire

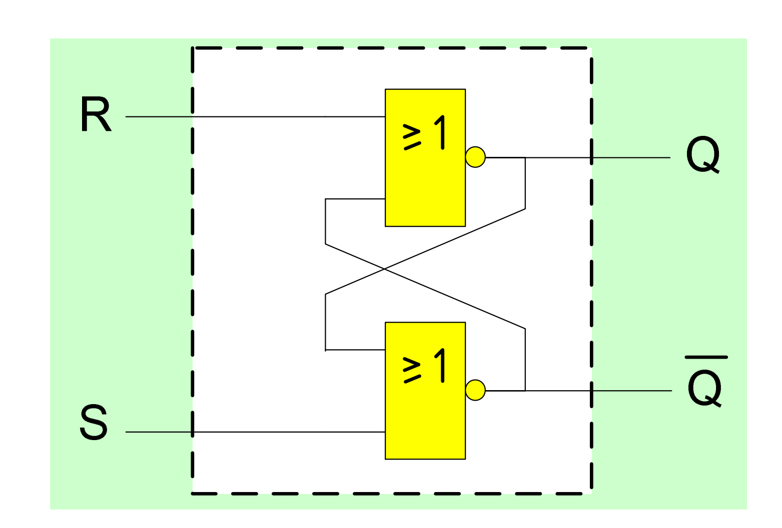
Les bascules et verrous logiques ont la propriété de mémoriser une information élémentaire (bit).

5-1-1- Verrou RS

Un verrou (***latch*** en anglais) RS a deux entrées (R et S) et deux sorties complémentaires Q et $\overline{Q}$

Symbole (fig. 10)

Logigramme (fig. 9)



Analyse du fonctionnement

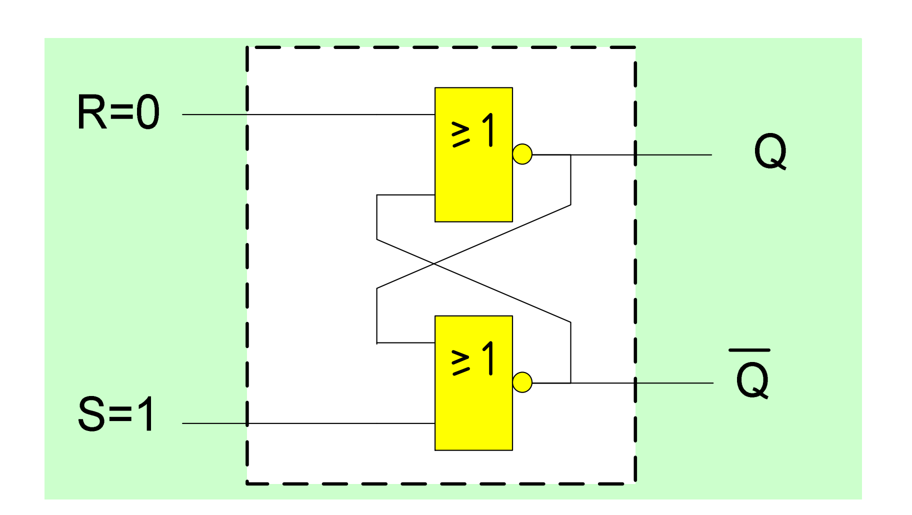
On s’interdit d’avoir en entrées (RS) = (11).

Quatre transitions possibles en entrées :

(RS) = (00) → (01)
(01) → (00)
(00) → (10)
(10) → (00)

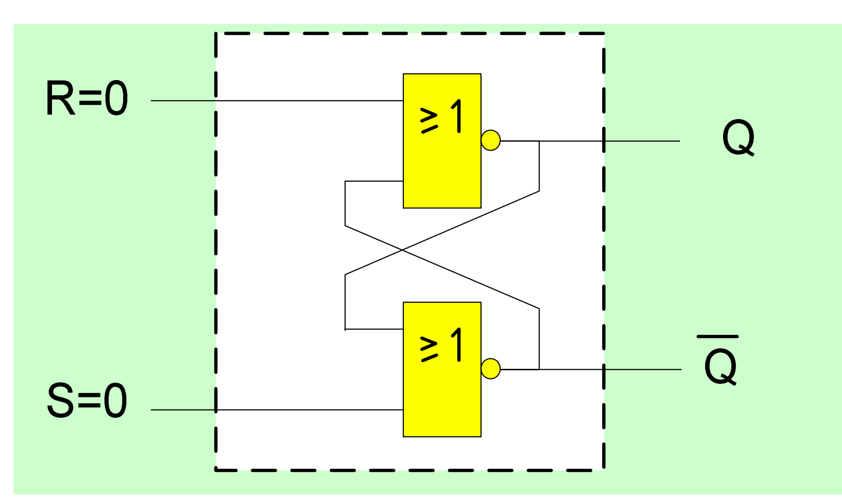
Rappel : la sortie d’une fonction NON OU est à l’état 1 si et seulement si toutes les entrées sont à l’état 0.

- Transition (00) → (01)
  (fig. 11a)



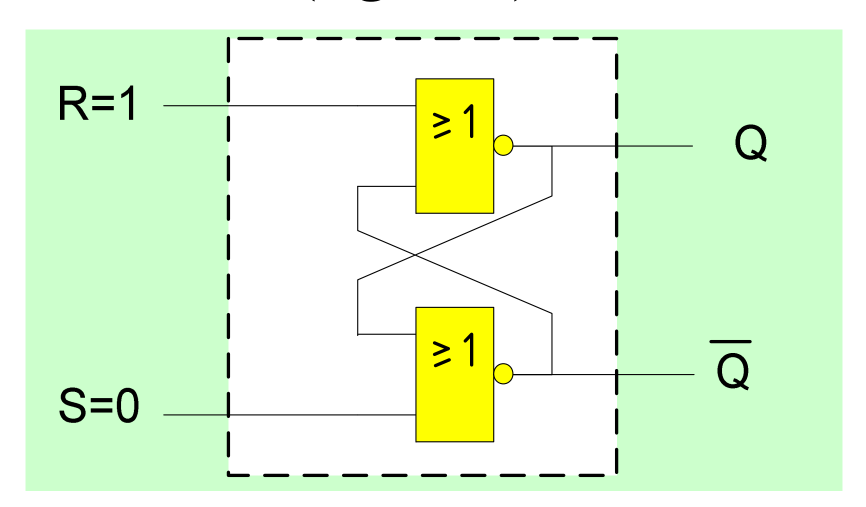
(RS) = (01) ⇒ Q = 1 : fonction **SET** (mise à 1 de la sortie Q).

- Transition (01) → (00)
  (fig. 11b)



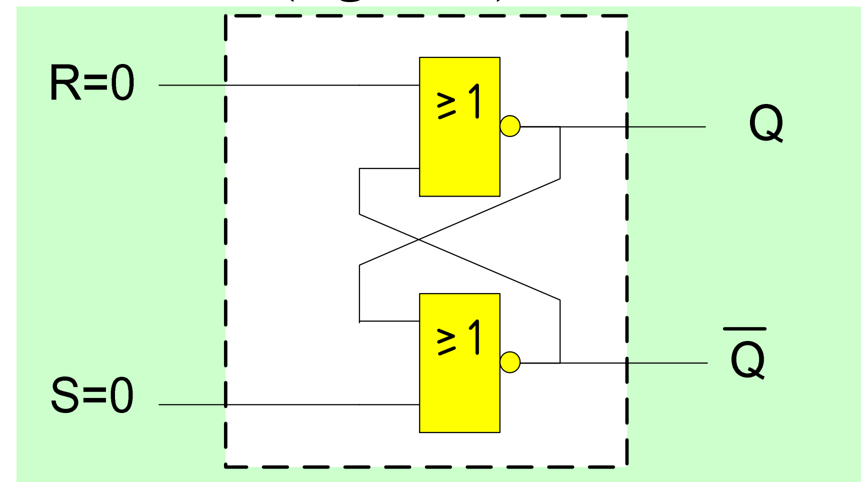
Les niveaux logiques en sortie sont inchangés :
fonction **mémoire**.

- Transition (00) → (10) (fig. 11c)



(RS) = (10) ⇒ Q = 0 : fonction **RESET** (mise à 0 de la sortie Q).

- Transition (10) → (00) (fig. 11d)

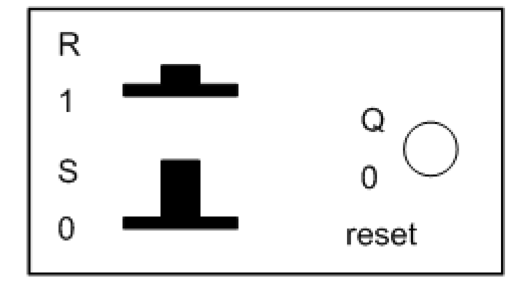


Fonction **mémoire**.

Table de vérité (table 16)

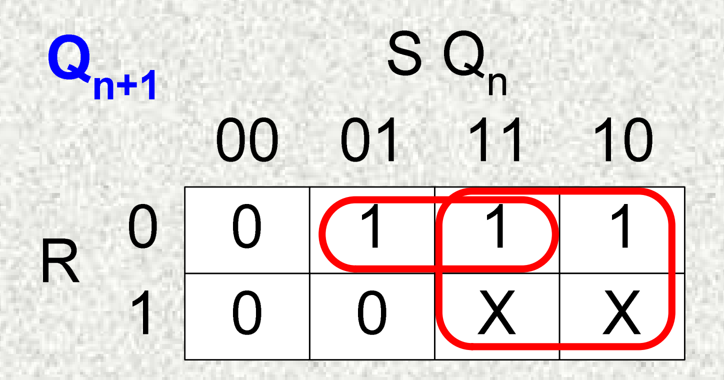
Notons Q_n et Q_{n+1} l'état de la sortie *avant* et *après* la transition en entrée :

R	S	Q_{n+1}	Fonction
0	0	Q_n	Mémoire
0	1	1	Set
1	0	0	Reset
1	1	X	Interdit



R	S	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	X
1	1	1	X

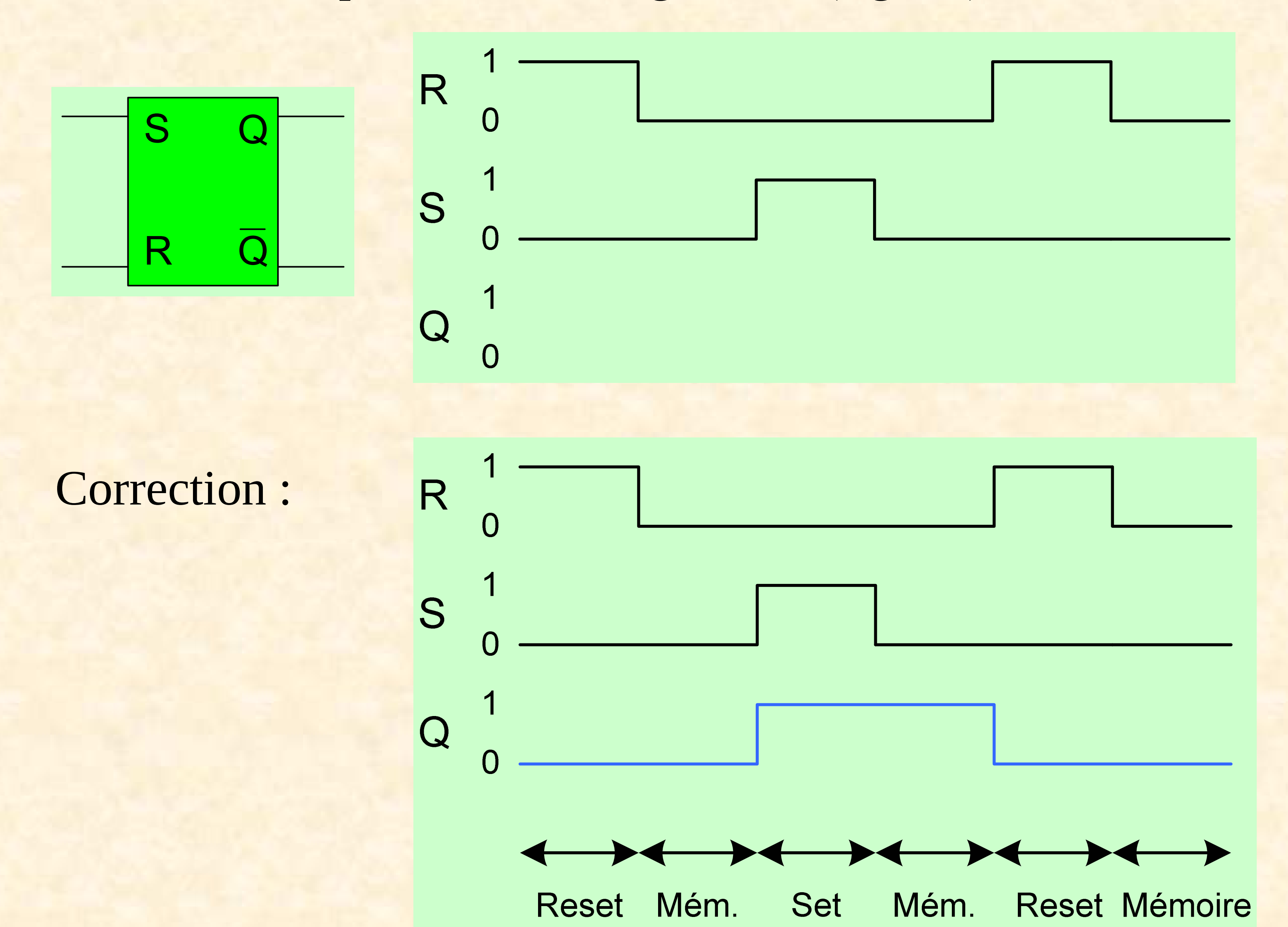
Tableau de Karnaugh



Equation logique :

$$Q_{n+1} = \overline{R} \cdot Q_n + S$$

Exercice : compléter le chronogramme (fig. 12)



5-1-2- Verrou RSH

Le verrou RSH est un verrou RS possédant une troisième entrée : H (horloge).

• quand H est active (niveau 1) : le verrou RSH se comporte comme un verrou RS.
• quand H est inactive (niveau 0) : verrouillage (fonction mémoire)

Symbole (fig. 13a)

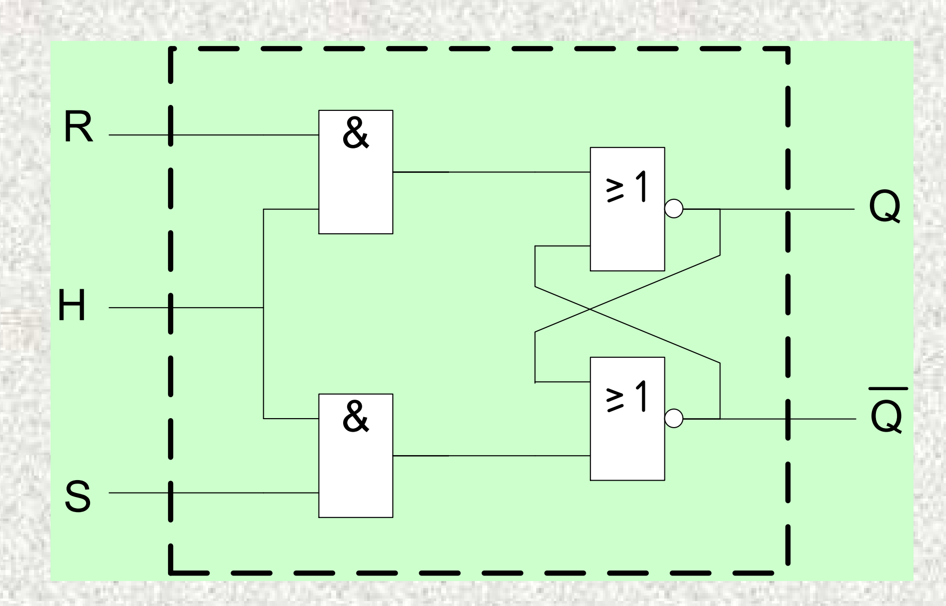
Table de vérité (table 17)

R	S	H	Q_{n+1}	Fonction
X	X	inactive	Q_n	Mémoire
0	0	active	Q_n	Mémoire
0	1	active	1	Set
1	0	active	0	Reset
1	1	active	X	Interdit

Equation logique :

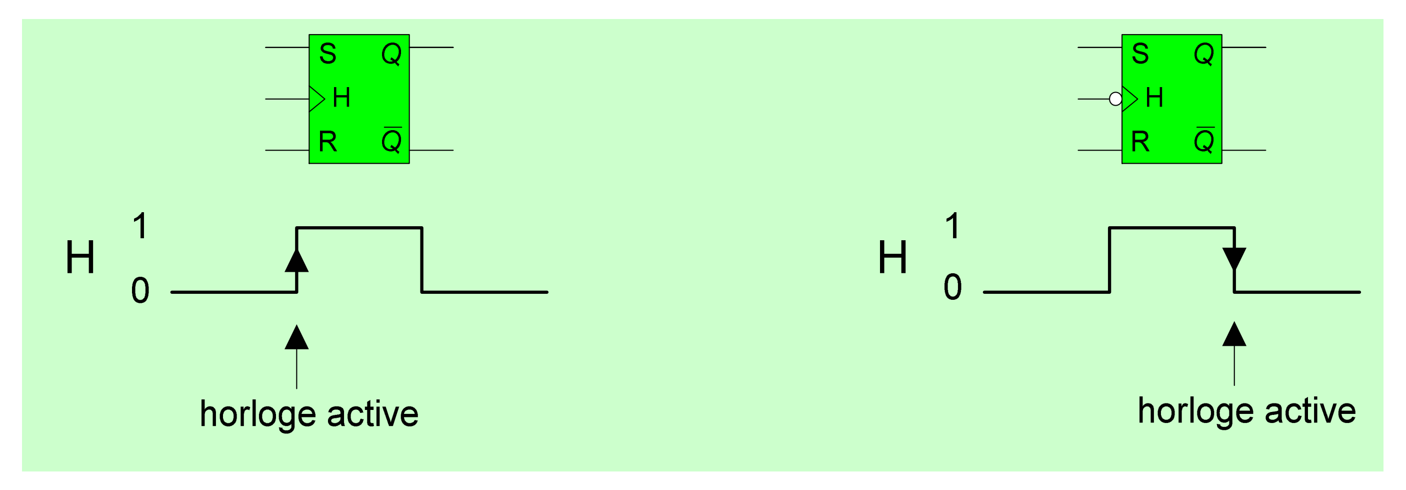
$$Q_{n+1} = H \cdot \left(\overline{R} \cdot Q_n \quad + \ S \right) + \overline{H} \cdot Q_n$$

Logigramme



5-1-3- Bascule RSH

Symboles (fig. 13b)



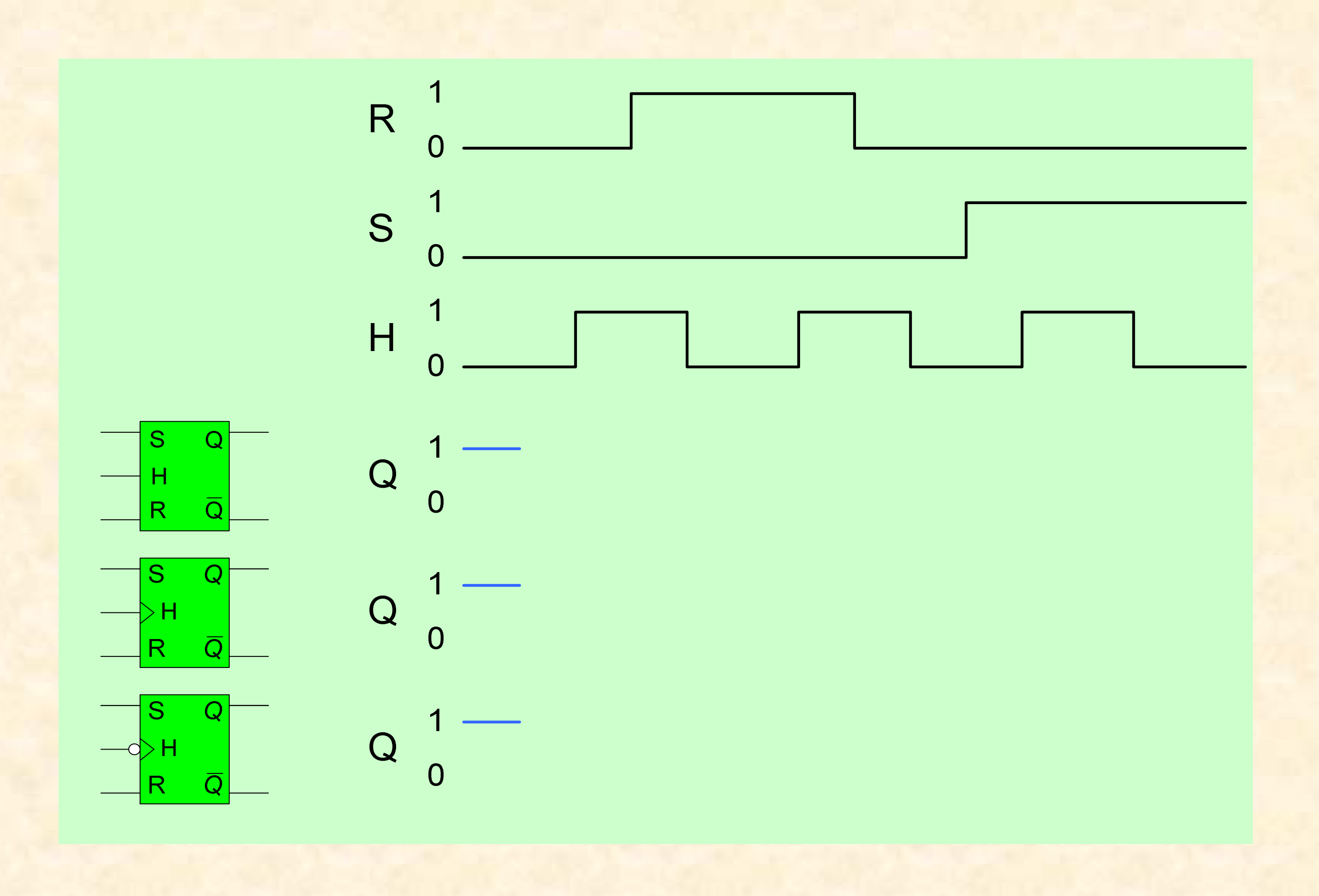
• deux types de bascule (***flip-flop*** en anglais) RSH :

- à horloge active :

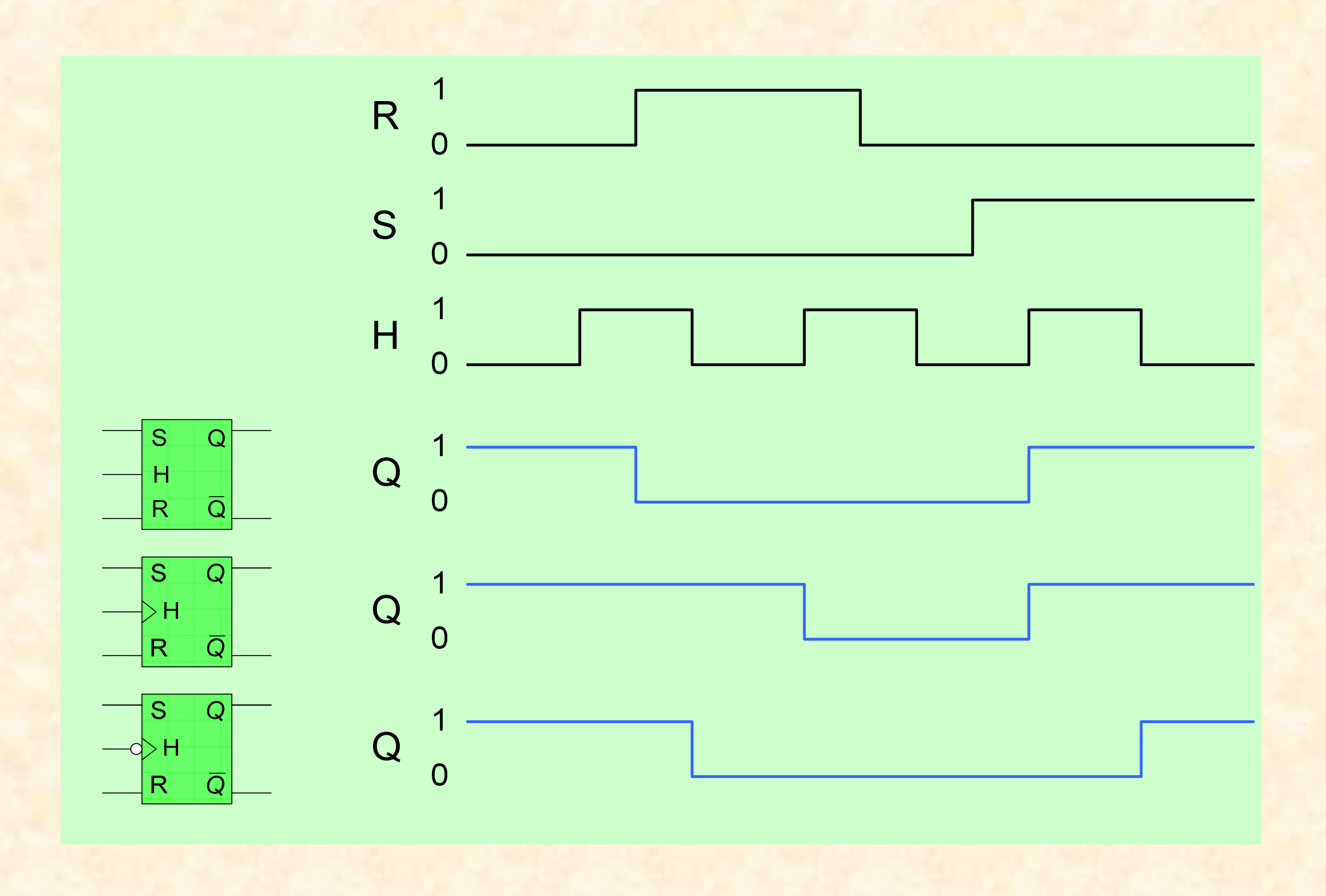
  - sur « front montant » (à l'instant où H bascule de 0 à 1)
  - sur « front descendant » (à l'instant où H bascule de 1 à 0)

• La table de vérite est la même que celle du verrou RSH.

Exercice : compléter les chronogrammes (fig. 14)



Correction :



5-1-4- Verrou D

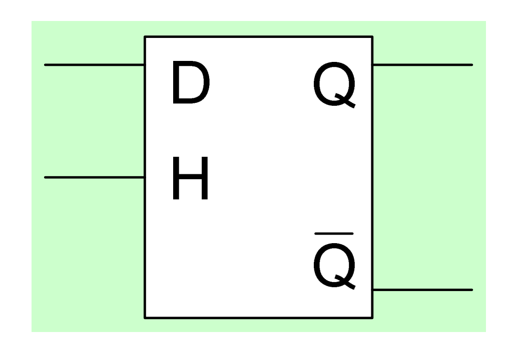


Table de vérité (table 18)

D	H	Q_{n+1}	Fonction
X	inactive	Q_n	Mémoire
0	active	0	Reset
1	active	1	Set

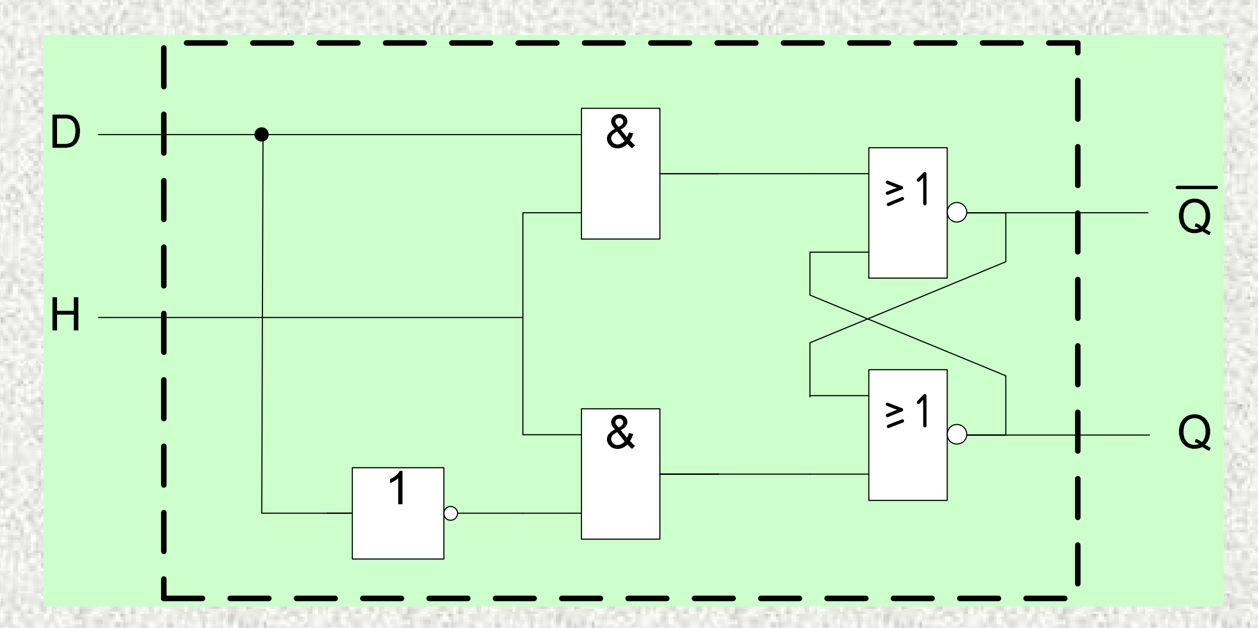
Lorsque l'horloge est active (niveau 1), le niveau présent à l'entrée D est transféré en sortie ($Q_{n+1} = D$).

Lorsque l'horloge est inactive (niveau 0), la sortie est « verrouillée ».

Equation logique :

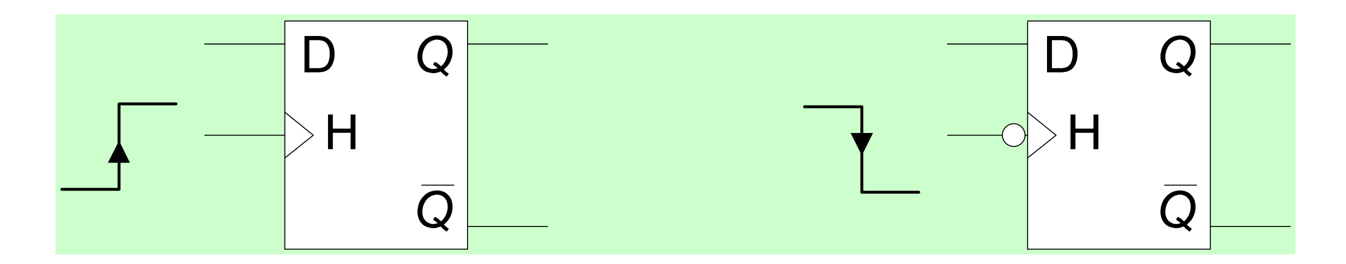
$$Q_{n+1} = H \cdot D + \overline{H} \cdot Q_n$$

Logigramme

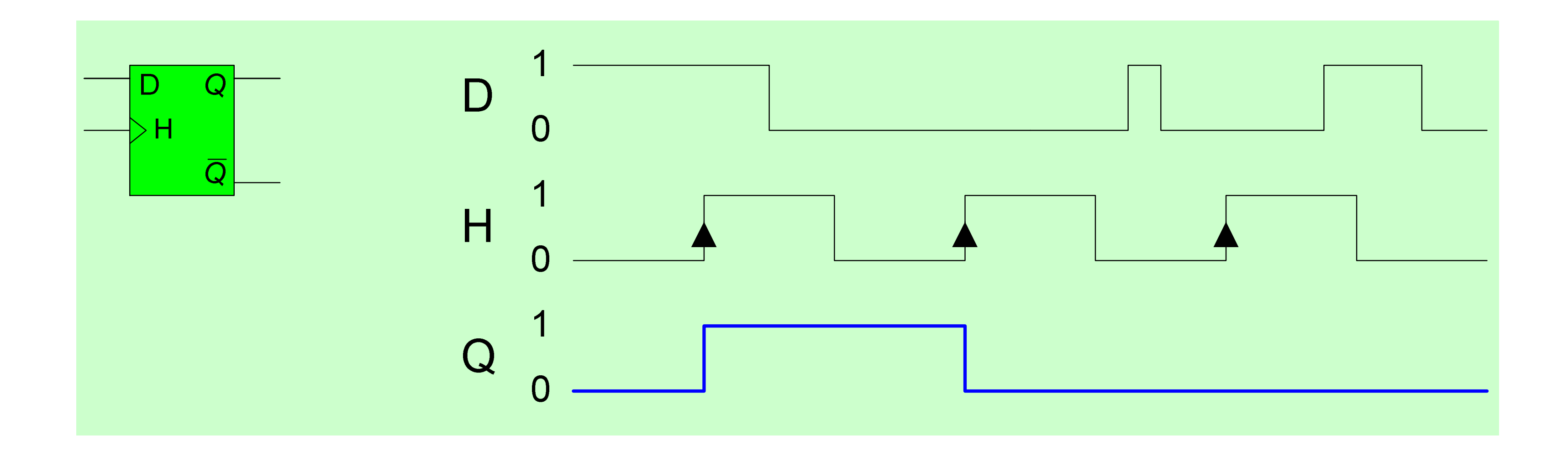


5-1-5- Bascule D

Symboles (fig. 15a)



Ex. de chronogrammes (fig. 15b)



5-1-6- Bascule JK

Symboles (fig. 16)

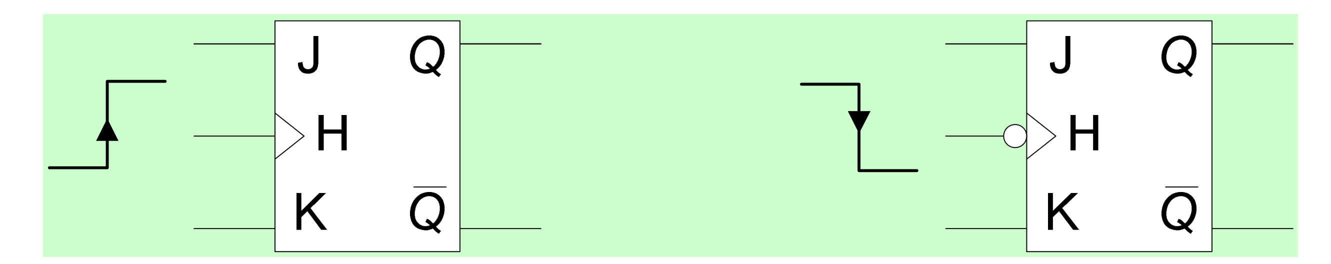


Table de vérité (table 19)

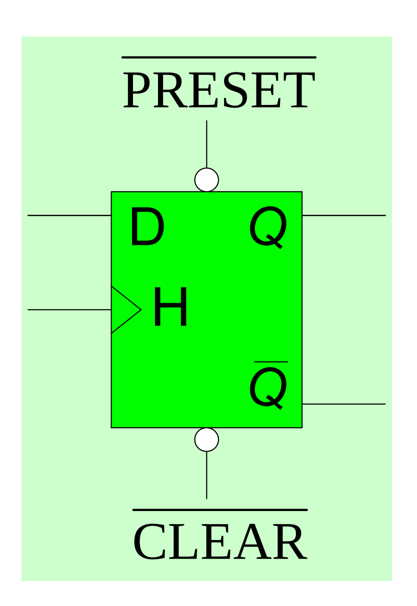
J	K	H	Q_{n+1}	Fonction
X	X	inactive	Q_n	Mémoire
0	0	active	Q_n	Mémoire
0	1	active	0	Reset
1	0	active	1	Set
1	1	active	$\overline{Q_n}$	*Basculement*

Equation logique :

$$Q_{n+1} = H \cdot \left(J \cdot \overline{Q_n} + \overline{K} \cdot Q_n \right) + \overline{H} \cdot Q_n$$

5-1-7- Entrées asynchrones

Exemple : 7474 (TTL)



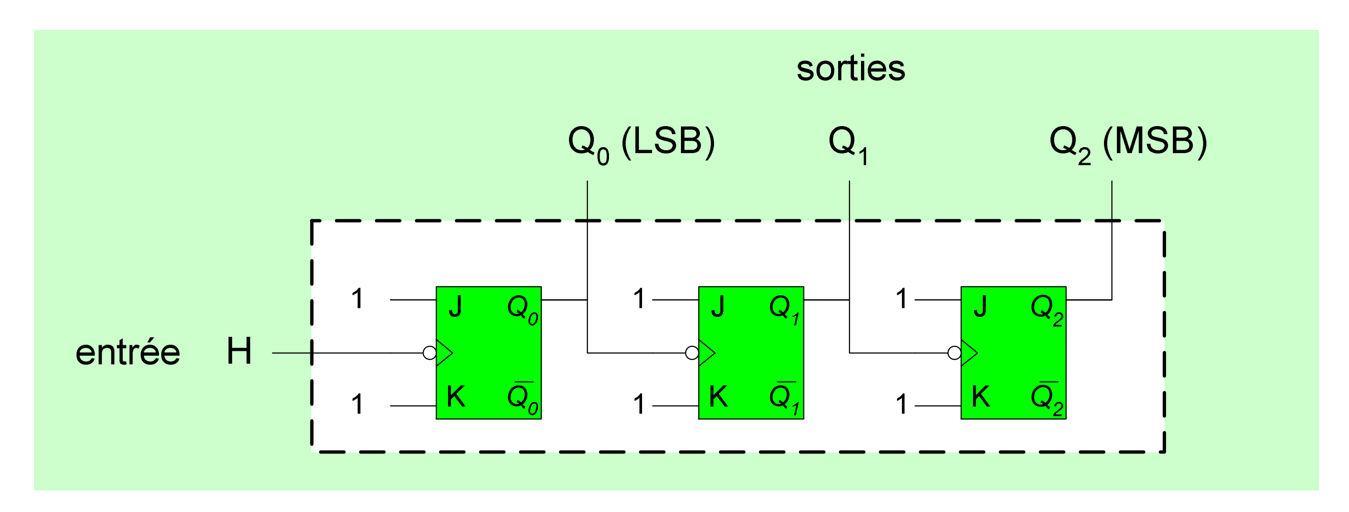
$$\overline{CLEAR} = 0 \quad \Rightarrow \quad Q = 0$$

$$\overline{PRESET} = 0 \quad \Rightarrow \quad Q = 1$$

5-2- Fonction comptage : les compteurs numériques

Le comptage nécessite la fonction mémoire :
on se sert donc de bascules.

• Exemple : Compteur binaire asynchrone 3 bits (fig. 17)



Ce circuit présente une entrée H et trois sorties.

Les sorties forment un nombre de 3 bits $(Q_2Q_1Q_0)_2$ qui donne le résultat du comptage.

Chronogrammes (fig. 18)

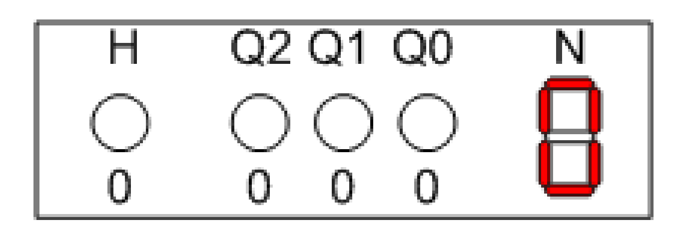
H 1 0

Q_0 1 0

Q_1 1 0

Q_2 1 0

$(Q_2Q_1Q_0)_2$	000	001	010	011	100	101	110	111	000	001	010
$(N)_{10}$	0	1	2	3	4	5	6	7	0	1	2



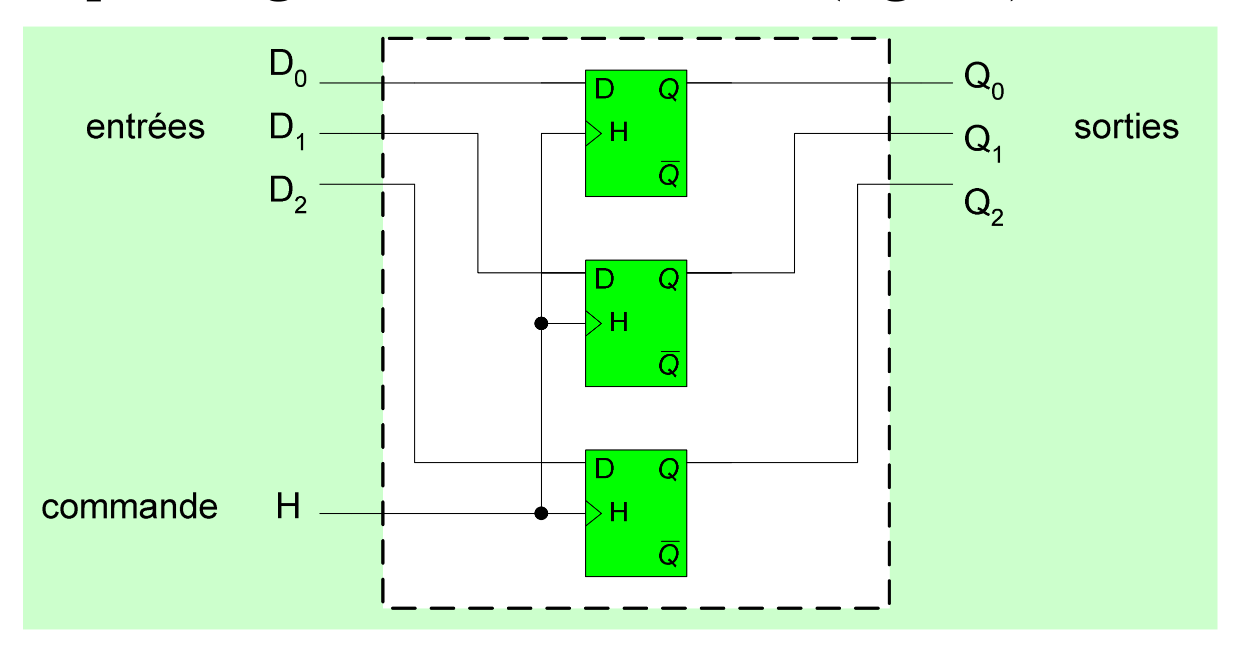
Remarque :

avec n bascules, on compte de 0 à $2^n - 1$ (compteur modulo 2^n).

5-3- Les registres

5-3-1- Registre mémoire

- Exemple : registre mémoire 3 bits (fig. 19)

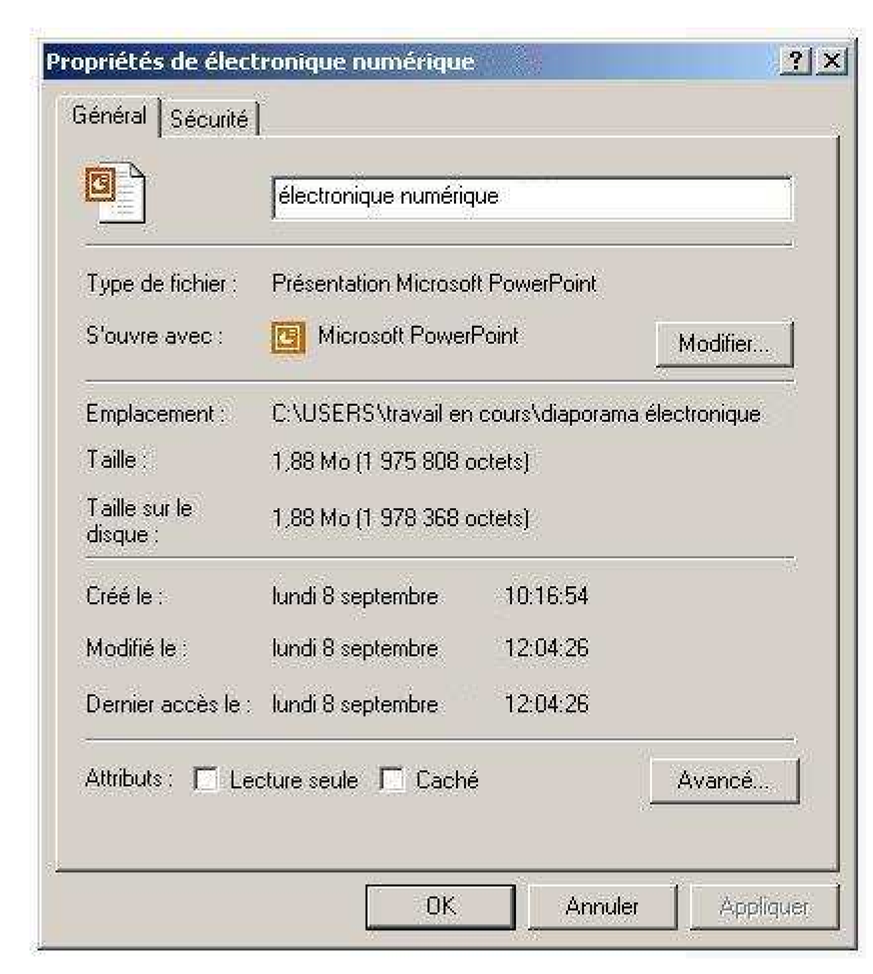


Quand l'horloge devient active, les trois bits présents en entrée sont transférés en sortie.
Les trois bits restent ensuite mémorisés en sortie aussi longtemps que l'horloge est inactive.

• Applications

- registres de microprocesseurs (32 bits)
- mémoire SRAM : mémoire cache des ordinateurs …

• Remarque : taille mémoire



1 octet =1 byte = 8 bits
1 « ko » = 2^{10} = 1024 octets
1 « Mo » = 1024 ko
= 1 048 576 octets
1 « Go » = 1024 Mo

A.N. une mémoire SRAM de 256 ko nécessite :

$256 \times 1024 \times 8$ = 2 097 152 bascules

5-3-2- Registre à décalage

• Exemple : registre à décalage 3 bits (fig. 20)

